Molecular Graph Augmentation with Rings and Functional Groups

Kurt De Grave* and Fabrizio Costa

Katholieke Universiteit Leuven,

Department of Computer Science, Celestijnenlaan 200A, 3001 Heverlee, Belgium

E-mail: kurt.degrave@cs.kuleuven.be

Abstract

Molecular graphs are a compact representation of molecules, but may be too concise to obtain optimal generalization performance from graph-based machine learning algorithms. Over centuries, chemists have learned what are the important functional groups in molecules. This knowledge is normally not manifest in molecular graphs. In this paper, we introduce a simple method to incorporate this type of background knowledge: we insert additional vertices with corresponding edges for each functional group and ring structure identified in the molecule. We present experimental evidence that, on a wide range of ligand-based tasks and data sets, the proposed augmentation method improves the predictive performance over several graph kernel based QSAR models. When the augmentation technique is used with the recent Pairwise Maximal Common Subgraphs Kernel, we achieve a significant improvement over the current state-of-the-art on the NCI-60 cancer data set in 28 out of 60 cell lines, with the other 32 cell lines showing no significant difference in accuracy. Finally, on the Bursi mutagenicity data set, we obtain near-optimal predictions.

*To whom correspondence should be addressed

Introduction

Virtual screening is an increasingly important component of the search for novel drug lead compounds.[1] There are two fundamental approaches: target-based and ligand-based. One can use docking to estimate affinity only if the 3D structure of the target protein is available in sufficient detail. By contrast, ligand-based screening does not require specific knowledge about the protein structure, but ranks or classifies molecules in a database according to their similarity to known active and inactive molecules. When an explicit model is built on a set of ligands with known activity levels, it is commonly called a quantitative structure-activity relationship (QSAR) model. A most critical aspect of accurate ligand classification is how to represent molecules for algorithmic processing.[2] In this paper, we describe a novel method for representing molecular graphs. A graph is a mathematical structure, consisting of a set of vertices (also called nodes) and a set of binary relations between vertices, called edges (or links). It is very natural to represent a molecule as a graph where atoms are vertices and bonds are edges. In fact, graph theory and chemistry advances are historically tightly linked.[3] Molecular graphs, when appropriately labeled, can be a lossless representation. I.e., as long as care is taken to encode stereogenic centers, a synthetic chemist could in principle reproduce the exact molecule from the information contained in a molecular graph. Molecular graphs are also a very compact and efficient representation. Common file formats for molecules, such as MDL mol and Sybyl mol2, essentially store an atom-bond graph.

Methods

Molecular graph augmentation

Motivation

Even though atom-bond graphs are a representation that is both compact and potentially lossless, they are not necessarily the best representation for a machine learning algorithm to achieve maximal predictive performance. We say that an algorithm has good generalisation capabilities when

it can predict a property, such as the level of a certain biological activity, of unseen instances, after learning from a limited number of observations. When human experts have to make such predictions, they access prior knowledge about the behavior of molecules that has been built up by generations of chemists. The knowledge has been gained by observing numerous phenomena in the past, directly or loosely related to the property under study. The value of the prior knowledge (or background knowledge) can be expressed as a number of "free" extra observations of the property under consideration. To see why, consider the equivalence class notion induced by the donor/acceptor property. Assume that the activity of a molecule remains unchanged when a donor is present in a specific position, then the set of molecules obtained by swapping functional groups that maintain the presence of the donor atom in that specific position are all equivalent. This equivalence property can be either encoded specifying (in a compact way) the notion of donor/acceptor or by explicitly enumerating all molecular variants that are equivalent. A computer algorithm, in contrast to the human expert, when presented molecules encoded in their non-redundant atom-bond representation, has access only to the direct observations. It is therefore natural to try to encode the prior knowledge of chemists, either as part of a special-purpose learning algorithm, or as part of the representation of the molecules as they are presented to the algorithm. We will take the latter option. Although the idea of altering the information encoded on the chemical graph is not novel by any means, we will take a novel augmentation approach: we start from the atom-bond graph representation and add background knowledge, more specifically we annotate functional groups and rings as they are described by most chemical textbooks, using extra vertices. In the remainder of the text we will use the general term moiety to refer to either a functional group or a small ring.

Adding moiety nodes to an atom-bond graph

It has long been established that chemistry can be explained by underlying laws of physics. Yet only recently it has become clear that the full and exact information about a molecule's behavior is contained in the electron density field.[4] However, for practical and computational purposes, the continuous, three-dimensional field must be discretized at some point. Functional groups are essen-

tially an ante litteram, empirically established, discrete set of electron density cloud characteristics that remain fairly constant for a definite group of atoms, independent of their environment.[5] Functional groups can hence be considered to be an information-dense discretizing approximation of molecular behavior.

DMax[6] is an Inductive Logic Programming[7] (ILP) system with specialized background knowledge to tackle chemical and biological problems. It is related to the ACE relational data mining system,[8] from which it inherited some components, such as the query refinement operator and the Prolog system. The version of DMax for QSAR rule induction is called DMax Chemistry Assistant (DCA). The program finds rules describing (potentially complex) substructures and properties of molecules that are positively or negatively correlated with the measured biological activity. For this purpose, the tool has a sophisticated built-in library to calculate functional groups and rings of a compound,[9] which it uses as building blocks for more complex rules. The identified moiety instances are stored in a special-purpose relational database. In this paper, we extract information from this database to construct augmented graphs. All 77 moieties for which nodes are added, are listed in Table 1.

DCA defines a hierarchy of moieties to be able to discover activity-correlated rules with the most appropriate specificity. For example, 'any amide group' is more general than 'sulfonamide'. If there is pertinent evidence in the observations, DCA can hypothesize that in a specific location the presence of a sulfonamide is critical (probably in addition to other requirements in nearby locations). However, if the data contains counterexamples that achieve a high level of activity without the sulfone, DCA will allocate more credence to the alternative hypothesis: that any type of amide group is sufficient. In this work, we make use only of the most specific moiety definitions. Since we do not explicitly add generalized functional group concepts such as 'any amide group' or 'any ether', we will rely entirely on the machine learning algorithm for generalisation. We also exclude composed concepts, such as phenol or urea. A possible extension of the method is to use additional nodes to represent more general concepts in the hierarchy, or encode them using extra labels.

Table 1: List of all moiety types introduced during augmentation.

benzene ring	thioamide
pyrrole ring	sulfonamide
furan ring	sulfinamide
thiophene ring	oxime
pyrazole ring	thioxime
imidazole ring	imine
pyridine ring	hydroxylamine
pyridazine ring	thiohydroxylamine
pyrimidine ring	amine
pyrazine ring	n-hydroxyamide
(other) hetero-aromatic ring	n-sulfanylamide
(other) non-hetero-aromatic ring	hydroxyammonium
hetero-non-aromatic ring	sulfanylammonium
non-hetero-non-aromatic ring	ammonium ion
methyl	nitroso
phosphate	thio-S-carboxylic ester
phosphonate	dithiocarboxylic ester
phosphinate	thioether
miscellaneous phosphor	thio-S-carboxylic acid
acylhalide	dithiocarboxylic acid
halide	thiol
carboxylic ester	conjugated base of a thio-S-carboxylic acid
thio-O-carboxylic ester	conjugated base of a dithiocarboxylic acid
methoxy	sulfide
ether	n-hydroxythioamide
carboxylic acid	n-sulfanylthioamide
thio-O-carboxylic acid	sulfoxide
alcohol	sulfinic acid
conjugated base of a carboxylic acid	sulfinic ester
conjugated base of a thio-O-carboxylic acid	conjugated base of a sulfinic acid
oxide	sulfonic acid
ketone	sulfonic ester
aldehyde	conjugated base of a sulfonic acid
diazo	sulfone
azide	metal ion
nitro	counterion
nitrile	(other) heteroatoms
iminium ion	aliphatic chain
amide	

It is important to stress the fact that simple subgraph matching is not sufficient to identify moieties, for two reasons:

- The matching may be context-sensitive. For example, a nitrogen is not considered an amine if it is connected to a carbonyl group. Instead, the atoms collectively act as an amide function.
- Some structural variation may be allowed, such as resonance structures or bioisosteres. For example, a hetero-non-aromatic ring may contain any non-carbon atom.

In DCA, functional group identification is implemented by means of logic programming.[10] Because of the above and the expressivity gap between Turing-complete logic programs and SMARTS, the precise definitions of the functional groups as perceived by DCA cannot in general be captured by concise SMARTS strings.

To elucidate the augmentation process, Figure 1 shows the composition of the augmented graph of ribavirin, a nucleoside antimetabolite antiviral agent. The basis is the atom-bond graph as defined by the molecule's structural formula, where atoms are vertices labeled with the atom type, and bonds are edges labeled as either single, double, triple, or aromatic. Hydrogen atoms are omitted. We used DCA's aromaticity perception. It verifies Hückel's rule for (systems of) 5- and 6-rings. In the case an O, N, or S atom is included in the ring between single bonds, the ring can also be aromatic.

To obtain the augmented graph, the following steps are performed:

1. Vertices for all moieties defined in the background knowledge are added to the atom-bond graph.
2. We add part-of edges between the moiety vertices and their constituent atoms. Atoms can be part of multiple moieties.
3. The moieties are linked with an edge labeled as: a) *fused* when their constituent atom level subgraphs share one or more vertices; or as b) *connected* when their constituent atom level subgraphs do not have any vertex in common, but there exists an edge connecting vertices belonging to the two different moieties.

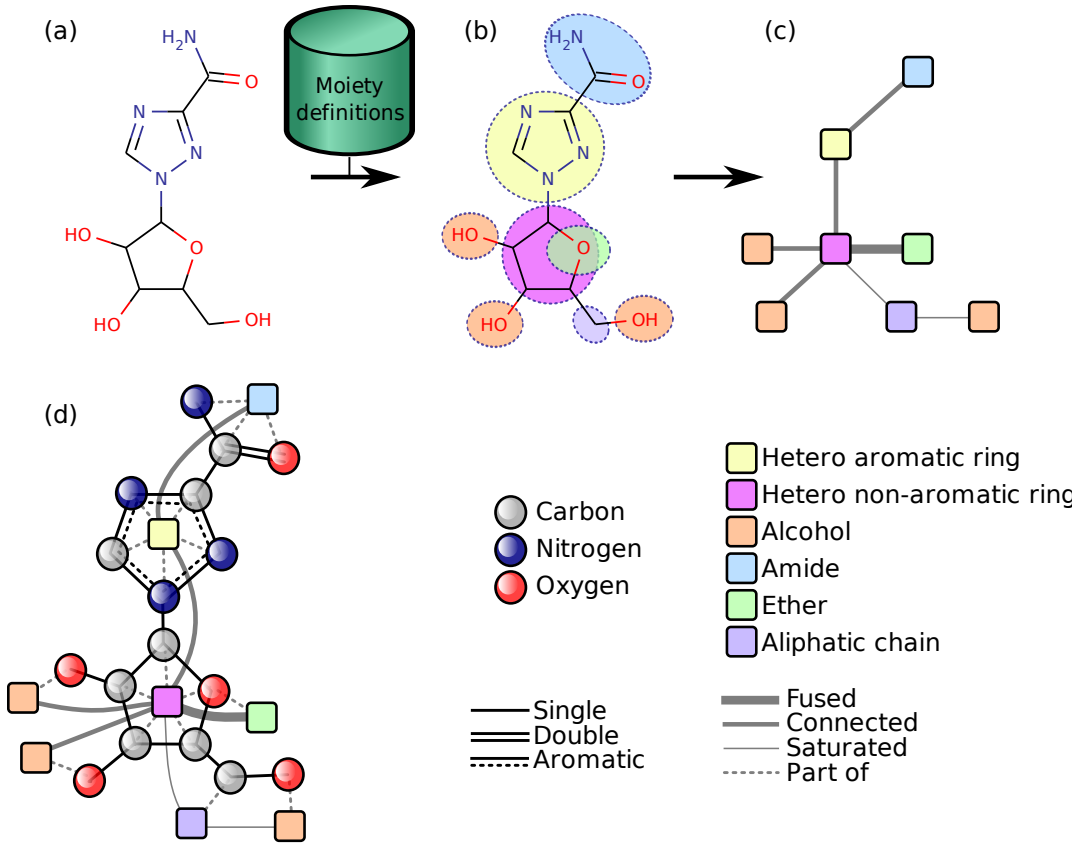

Figure 1: Molecular graph augmentation: (a) structure of ribavirin (input), (b) identification of functional groups and rings, (c) the moieties are encoded as extra nodes in the graph, which are added to the original structure to obtain (d) the augmented graph.

4. Edges connecting any moiety to an aliphatic chain are not labeled as *connected* but as either c) *saturated* if the chain is saturated, or d) *unsaturated* otherwise.

In the ribavirin illustration, the 1,2,4-triazole is not in the list of defined groups and is therefore represented by a vertex labeled as general hetero-aromatic ring.

As a result of the augmentation, the graph kernels have access to a large subset of the BK-Level-1 background knowledge from Ando et al.,[6] which primarily consists of the definitions of moieties and their relationships.

Finally, note that the method we present makes use of 2D information only; that is, we do not label stereogenic centers in any way and any model based on our representation is therefore unable to distinguish between stereoisomers.

Related molecular graph representations

Several alternative graph representations of molecules have been described for use in virtual screening. Previous approaches tended to use leaner, more abstract representations rather than an enriched one. A prime example are reduced graphs, first introduced by Gillet et al. for substructure searching and later adapted for similarity searching.[11–13] The key idea is to omit irrelevant details from the molecule and retain a more abstract graph. A node in the reduced graph may represent multiple atoms in the original graph. Several abstraction types are used, giving rise to different reduced graph types. For example, each ring may be reduced to a single node labeled R, a set of recognized functional groups may indiscriminatively be represented as nodes labeled F, and all other atoms can be mapped to catch-all link nodes if they separate the rings and features, giving rise to a Ring/Feature reduced graph with just three node types. The most detailed level of abstraction for functional groups in Gillet et al.'s 2003 paper[12] are hydrogen donors and acceptors. Reduced graphs allow to find more structurally dissimilar molecules than virtual screening by fingerprinting standard molecular graphs. In the process, unfortunately, most implementations sacrifice some generalisation performance. Stiefl et al.[14] introduced a variant called extended reduced graphs (ErG) with better generalisation performance than standard Daylight fingerprints.

The most similar approach to the method proposed here may be the one by Takahashi et al.[11] in 1992. There, the authors compute a graph of specific functional groups, unlike later work where more abstract concepts have been used. The vertices are labeled with topological distances (multiple if there are different paths).

Rarey and Dixon[15] proposed to represent molecules by feature trees. A molecule graph is converted to a tree by iteratively collapsing its minimal-length cycles into single nodes. Each node in the tree is associated with a vector of chemical features, such as approximated Van Der Waals volume and hydrogen donorship. The features for larger subtrees can be computed from the features of its parts, e.g. by summation. A similarity function must be provided for each feature. The direct similarity of two subtrees is computed as the weighted sum of the similarity of their features. For comparing two molecules, the algorithm tries to find a coordinated way of splitting the molecules

in subtrees such that the highest aggregate direct similarity of matched subtrees is obtained. The tree representation is a theoretically attractive middle ground in between graphs, where most operators are computationally expensive, and vectors, which lack expressiveness. As opposed to the feature trees algorithm, the approach taken in the current work separates the representational concern from the similarity computation. This allows to delegate the computation of the similarity to a graph kernel, which is positive semi-definite, a property so far not attributed to feature tree similarity scores. A kernel, unlike a similarity score, allows the direct use of kernel-based machine learning models, such as support vector machines.

Graph kernels

In this paper we investigate the effect of graph augmentation on the QSAR modeling performance of kernel-based classifiers. Kernel methods have proved to be able to achieve state-of-the-art performance in many classification tasks. Due to their versatility, they can be employed in domains where the instances are more conveniently represented in a structured form, such as sequences and graphs. This property makes them the ideal choice to tackle bioinformatics and chemoinformatics tasks, where DNA sequences are naturally represented as linear chains and molecules are represented as graphs.

The main idea in the kernel approach is to devise a computationally efficient way to calculate the similarity between two instances[1] and to cast the classification problem into a convex optimization problem. In this way we are guaranteed the existence of a globally optimal solution (that is, there is no risk to obtain approximate solutions that have just reached local optima, as in the case of neural network techniques) and the practitioner can tap into a vast literature on optimization solvers and advanced techniques to speed up the computation. For a reference on the kernel approach in Machine Learning, see Cristianini and Shawe-Taylor [16] or Schölkopf and Smola.[17]

Here, we are concerned with small molecule tasks, therefore we look into kernel methods for

[1] More precisely, we want to compute the dot product of the instances once these are mapped in a (usually implicit) vector space of very high dimensionality.

graphs, for which an increasingly large literature exists (see Gärtner [18] for references). In this work we compare three different types of graph kernels: the Equal Length Shortest-Path Kernel, the Weighted Decomposition Kernel and the Pairwise Maximal Common Subgraphs Kernel. The choice is motivated by the desire to sample diverse approaches within the graph kernel techniques: the Equal Length Shortest-Path Kernel considers long distance interactions between pairs of vertices in a graph, the Weighted Decomposition Kernel considers the local information in the neighborhood of the vertices in a graph, while, finally, the Pairwise Maximal Common Subgraphs Kernel considers the occurrence of shared, very large subgraphs.

Equal Length Shortest-Path Kernel

Borgwardt and Kriegel [19] presented an efficient graph kernel based on shortest paths. In this work we use our own implementation of a specialization of the equal length shortest-path kernel (ELSPK). The idea is to compute the similarity between two graphs by comparing all the respective pairs of vertices annotated with their topological distances. This is achieved by 1) first calculating the shortest path distance between all pairs of vertices using Floyd-Warshall's algorithm,[20,21] and 2) then computing an all-pairs-shortest-paths kernel on edge walks of length 1 on an appropriately modified graph.

Formally, a graph G is transformed into a graph S such that there exists an edge between two nodes in S if they are connected by a path in G (i.e. S is the complete graph of the vertex set of G when G is connected). Every edge in S is labeled by the shortest distance between these two nodes, and we denote it with the term *"distance-edge"*. Given the vertex set $\mathbb{V}_S$ and edge set $\mathbb{E}_S$, the ELSPK is defined as:

$$K(S,S') = \sum_{e \in \mathbb{E}_S} \sum_{e' \in \mathbb{E}'_S} k^{(1)}(e,e') \tag{1}$$

where $k^{(1)}$ is a positive semi-definite kernel on edge walks of length 1.

As $k^{(1)}$, we consider the exact matching kernel over edges, where two edges match if they have

the same label and if the labels of their vertices also match:

$$k^{(1)}(e,e') = \delta(e,e') = \begin{cases} 1 & \text{if } \mathscr{L}(v) = \mathscr{L}(v') \text{ and } \mathscr{L}(u) = \mathscr{L}(u') \text{ and } \mathscr{L}(e) = \mathscr{L}(e') \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $e = (v,u)$, $\mathscr{L}(v)$ is the vertex label, and $\mathscr{L}(e)$ is the edge label, i.e. in this case the topological distance between v and u in G.

Finally, in this work we specialize $k^{(1)}$ to its zero extension parameterized by a maximum distance d, that is, we consider:

$$k_d^{(1)}(e,e') = \begin{cases} k^{(1)}(e,e') & \text{if } \mathscr{L}(e) = \mathscr{L}(e') \leq d \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In words: we consider all pairs of vertices up to a maximum distance d and count how many exact matches we have between the distance-edge sets representing the two original graphs. In Figure 2 we give a graphical representation of the distance-edge set induced considering a given single vertex.

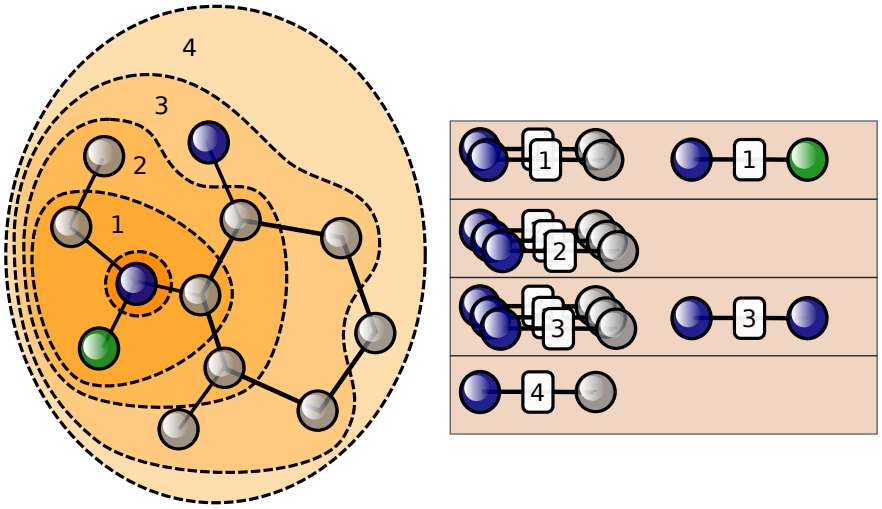

Figure 2: The set of distance-edges induced by a single vertex for the ELSPK. On the left, the original graph G is depicted. The table on the right shows only the descriptors induced by the vertex of interest, the highlighted nitrogen. The kernel will complete the table for all other vertices. Two molecules are then compared by counting the elements in the intersection of their respective tables.

Weighted Decomposition Kernel

The weighted decomposition kernel (WDK) is a specialization of a decomposition kernel,[22] introduced by Menchetti et al. [23] The idea is to compare a large *context* associated to each atom. More precisely, each vertex v in a graph G is characterized by a small region of topological *nearby* elements – the context $\mathscr{C}^l_G(v)$ – defined as the subgraph composed of the vertices within distance l from vertex v. The similarity between two graphs is then computed in terms of the similarity of the set of their vertices $\mathbb{V}_x$ weighted by the similarity of the their respective contexts.

Formally:

$$K(G,G') = \sum_{v \in \mathbb{V}_G} \sum_{v' \in \mathbb{V}_{G'}} \delta(v,v') \cdot k^{(l)}(v,v') \tag{4}$$

where the similarity between two vertices v and v' is computed by the exact matching kernel:

$$\delta(v,v') = \begin{cases} 1 & \text{if } \mathscr{L}(v) = \mathscr{L}(v') \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\mathscr{L}(v)$ is again the vertex label.

The context kernel $k^{(l)}$ is a set kernel computed over the context edges:

$$k^{(l)}(v,v') = \sum_{e \in \mathscr{C}^l_G(v)} \sum_{e' \in \mathscr{C}^l_{G'}(v')} \delta(e,e') \tag{6}$$

In words: the similarity of two node contexts is defined as the number of exact matches between the edges present in the contexts of v and v'. In Figure 3 we give a graphical representation for the context edge set of two vertices.

Pairwise Maximal Common Subgraphs Kernel

The Pairwise Maximal Common Subgraphs Kernel (PMCSK) introduced by Schietgat et al. [24] is a kernel built over structural keys that is computed in two steps: at first a set of relevant subgraphs is extracted from all possible pairs of instances; the subgraphs are then used to provide a bit vector

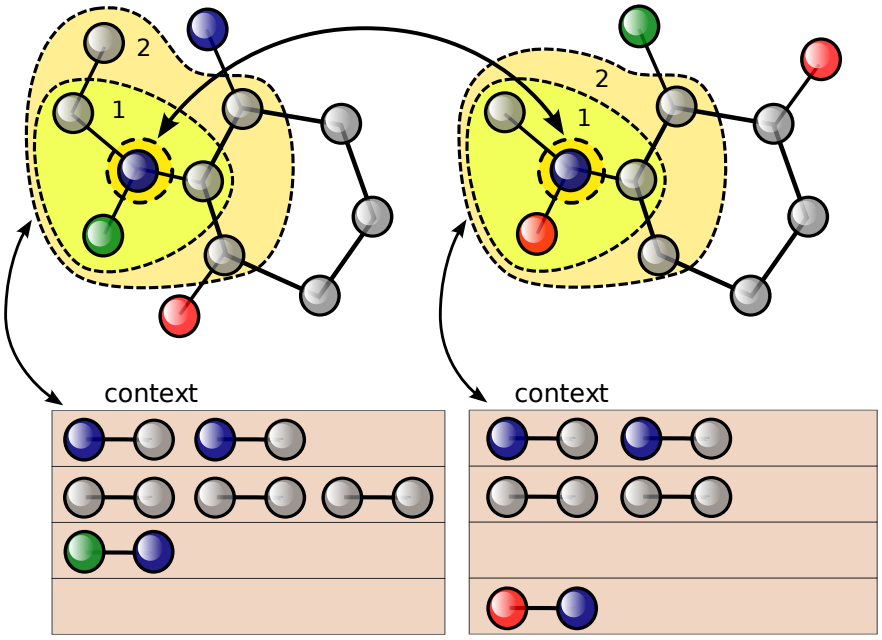

Figure 3: Context edge set for two vertices according to WDK with context radius $l = 2$. The two highlighted nitrogens can contribute to the similarity of the molecules because their atom type matches. How much they contribute, depends on the context they occur in: it is the number of exact matching edges in their respective contexts, which is 4.

encoding for a graph in the following way: the bit at position i is asserted if the i-th subgraph is present in the graph; finally the similarity between two graphs is computed via a Tanimoto kernel. The Tanimoto kernel (considered state-of-the-art for the classification of small molecules[25]) computes a similarity score by counting the number of common elements (i.e. the set-intersection) between the two instances as a fraction of the total number of elements that occur in both instances (i.e. the set-union). Formally, if an instance x has $|x|$ bits asserted, x' has $|x'|$ bits asserted and they share $|x \wedge x'|$ asserted bits, then the Tanimoto similarity score is a real number in the interval $[0,1]$ computed as:

$$\frac{|x \wedge x'|}{|x| + |x'| - |x \wedge x'|}.$$

Gower [26] shows that this similarity score is positive semi-definite, hence a proper Mercer kernel.

The novelty of the PMCSK lies in the definition of the relevant/interesting subgraph: a subgraph is relevant if it is the maximal common subgraph between two instances belonging to the

data set. This procedure differs from the usual structural keys building approaches in that there is no pre-defined dictionary of fragments nor it is considered the set of *all* (syntactically correct) fragments up to a pre-defined maximum size.

Computing the maximal common subgraph in the general case is an NP-hard problem. Fortunately, there exist a polynomial-time algorithm if one considers only outerplanar graphs in combination with the block-and-bridge-preserving (BBP) subgraph isomorphism.[27] Intuitively, a graph is outerplanar when it can be embedded in the plane in such a way that all of their vertices lie on the outside of the graph. In Figure 4 a) we give an example of a non-outerplanar molecule: here the graph cannot be drawn on the plane in such a way that the highlighted vertex is reachable from the outside of the graph. Since the vast majority ($\approx$90% of the molecules for each of the four data sets used in this paper) of small molecule graphs are outerplanar, this restriction does not represent a severe limitation in practice. While non-outerplanar graphs do not contribute to the identification of the MCSs, the presence of the MCSs extracted from pairs of outerplanar graphs is still used to build their vector encodings.

Finally, the BBP subgraph isomorphism is a special case of the general subgraph isomorphism. In BBP isomorphism we distinguish special subgraph configurations called *blocks* and *bridges*. Intuitively, a block is a ring-like structure (cycle) and a bridge is a structure that is not a block, such as linear sequence or a tree-like branching structure. The BBP isomorphism prescribes that only edges of bridges of a graph G can be mapped to edges of bridges of the other graph G' and edges of blocks of G can be mapped only to edges of blocks of G'.

In Figure 4 b) and c) we give an example of the consequences on the identification of a maximal common subgraph under the general notion of subgraph isomorphism and under the BBP notion.

An additional point in favor of the PMCSK is that the block-and-bridge-preserving subgraph isomorphism seems to produce better quality subgraphs for several chemoinformatics tasks when compared with the general subgraph isomorphism; that is, it has been experimentally observed that the induced predictive models exhibit better performances[24] when the subgraphs are extracted under the BBP isomorphism.

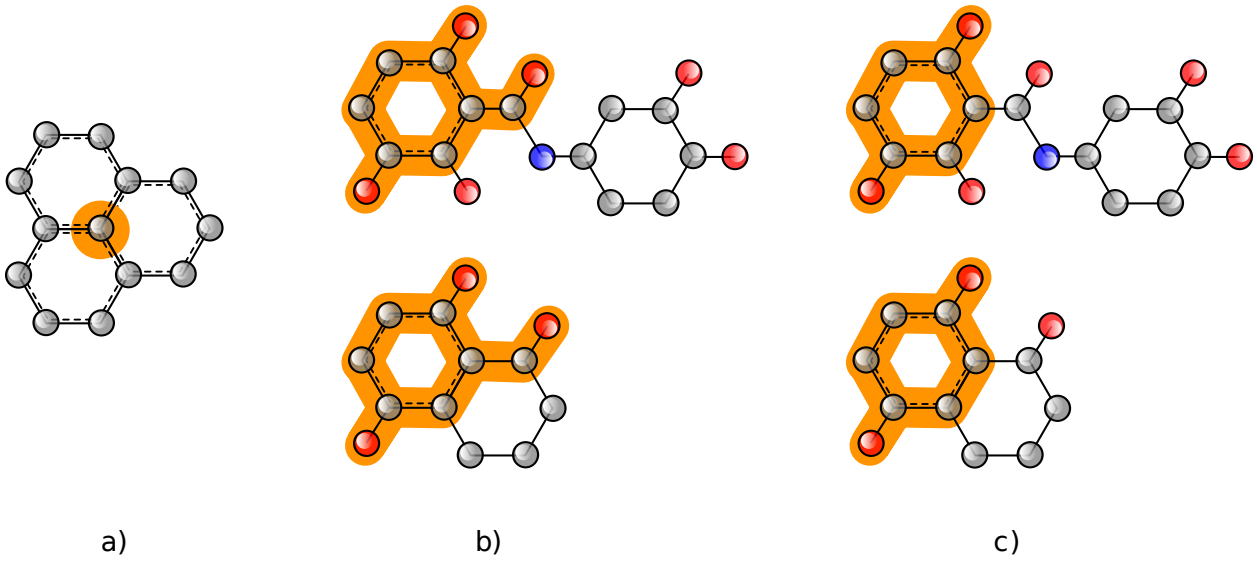

Figure 4: (a) Example of a non-outerplanar graph (b) a maximal common subgraph under general subgraph isomorphism (c) a maximal common subgraph under BBP subgraph isomorphism

Experimental Setup

Data sets

To determine the utility of the augmentation method, we we selected four data sets from three different domains: oncology, virology, and toxicology (in vivo and in vitro).

NCI-60

The Developmental Therapeutics Program (DTP) at the U.S. National Cancer Institute (NCI) has checked a large number of compounds for evidence of the ability to inhibit the growth of human tumor cell lines[2]. The roughly balanced subset used by Swamidass et al. [28] has become a popular benchmark for QSAR algorithm research, often referred to as either NCI-60 or just NCI. The data set contains growth inhibition measurements on 60 cell lines. Each cell line has inhibition data on about 3500 compounds. There are 3910 compounds in the set in total.

[2] http://dtp.nci.nih.gov/docs/cancer/cancer_data.html

HIV

The DTP also runs an AIDS antiviral screening, which has checked a large number of compounds for evidence of protection against HIV-1. The October 1999 release of the database[3] contains the structures of 42687 molecules. Each of the compounds was tested twice, and 422 were confirmed to be active (CA), 1081 are moderately active (CM), and 41184 are inactive (CI). Sometimes, Kramer's subset of 41768 molecules[29] is used to benchmark machine learning algorithms. The full set has also been used before, e.g. by Ceroni et al. [30] There are three binary classification tasks commonly considered for this data set: distinguishing between CA and CM, between CA+CM and CI, and between CA and CI.

PTC

The 2000-2001 Predictive Toxicology Challenge (PTC)[31] was devised to stimulate the development of machine learning techniques for predictive toxicology models. The data originates from the US National Toxicology Program (NTP). The training and test sets have a different class distribution and a different prevailing mode of action,[32] therefore we only use the (corrected) training set, which contains 417 molecules. The aim is to predict the carcinogenicity of the compounds in different rodents, in particular male mice (MM), female mice (FM), male rats (MR), and female rats (FR).

Bursi

Kazius et al. [33] have constructed a data set of 4337 molecular structures with corresponding Ames data[4]. Ames is a short-term in vitro assay designed to detect genetic damage caused by chemicals and has become the standard test to determine mutagenicity. The distribution is 2401 mutagens and 1936 nonmutagens.

[3]The data can be downloaded from `http://dtp.nci.nih.gov/docs/aids/aids_data.html`.
[4]Available at `http://www.cheminformatics.org/datasets/bursi/`

Evaluation measures

We will evaluate the generalisation performance of the kernels and the augmentation method by the area under the receiver operating characteristic (AUROC).[34] The ROC is the plot of the fraction of true positives versus the fraction of false positives. An AUROC score of 100% indicates perfect separation of positives from negatives, whereas a score of 0% indicates that all negatives were selected before the first positive. An algorithm that predicts a random order has an expected AUROC of 50%.

We also report the ROC_{50} score,[34] which is the area under the ROC up to the first 50 false positives[5]. The ROC_{50} provides similar insights as *lift* graphs and gives an idea of how reliable the predictions that the method considers most trustworthy effectively are. A ROC_{50} score of 100% again indicates perfect separation of positives from negatives, whereas a score of 0% indicates that none of the top 50 molecules selected by the algorithm were true positives. The expected ROC_{50} of a random prediction algorithm depends on the number of negatives N in the test set:

$$\mathbb{E}_{randompred}(ROC_{50}) = \begin{cases} 25/N & \text{if } N \geqslant 50 \\ 50\% & \text{otherwise} \end{cases} \tag{7}$$

Experimental goal and setup

We are now well equipped to perform experiments to answer two questions about the proposed methods:

Q1 Does the augmentation of molecular atom-bond graphs with moieties improve the predictive performance of graph kernels in a support vector machine?

Q2 How do augmented graph kernels compare to the current QSAR state-of-the-art?

We tested the predictive performance of the SVM-Light[35] support vector machine implementation by 10-fold cross-validation. The folds were always stratified and identical for all methods.

SVM-Light was run with all parameters at their default value except for the cost factor (-j),

[5]In order to avoid the size of the data set having an excessive impact on the score, the horizontal axis is rescaled such that a value of 1.0 on the axis corresponds to the 50th false positive.

which was set to the prevalence ratio of negative to positive examples as suggested by Morik et al..[36] Each of the kernels was normalized such that $K(G,G) = 1$.

In the following we report method specific observations.

ELSPK

The maximum distance parameter d has been optimized by inner cross-validation for each training fold of the Bursi data set. The range considered was 4, 6, 8, ... 26. For the other data sets, we used the value that was selected most frequently in the Bursi data set: 14 for atom-bond graphs and 24 for augmented graphs.

WDK

We used a context radius l of 4 for the WDK, motivated by the results on the HIV data set by Menchetti et al. [23]

In this paper, in order to have a clearer comparison, we employ a simpler setup than in Menchetti et al. [23] In particular we 1) do not compound the kernel with a Gaussian kernel, 2) we do not use the information on the partial charge over single atoms, and 3) we do not use the information about the context complement (that is we do not weigh the similarity of two nodes by considering the edge set similarity of the edges that are *not* part of the vertex context). In this way the analysis of the advantages and disadvantages of different methods can be compared on a clearer and fairer basis since the only information used stems from the atom and bond types.

PMCSK

Unfortunately the molecular graph augmentation procedure as detailed above cannot be directly used with the PMCSK since the presence of the *part-of* edges in the augmented graphs make them non-outerplanar. To circumvent this difficulty, we operated as follows: the MCS descriptor generation method was run separately on the atom-bond graphs and the moiety graphs, both of which are outerplanar on their own for the vast majority of drug-like molecules. More specifically

we observe that 90-92% of the atom-bond graphs are outerplanar in all data sets considered and that the fraction of the moiety graphs that are outerplanar ranges from 73% in the HIV data set to 91% in the PTC data set.

Finally, the bit-vector representations for the atom-bond graphs and the moiety graphs are concatenated in order to obtain the overall joint representation. We will use the notation PMCSK(G'_{aug}) for this approach.

To reduce the runtime for the HIV data set, only the relatively small set of "positive" molecules (either CA or CA+CM) were used to derive maximal common subgraphs.

Results and Discussion

Q1 *Does the augmentation of molecular atom-bond graphs with moieties improve the predictive performance of graph kernels in a support vector machine?*

Table 2 shows the average per-fold cross-validated AUROC and ROC_{50} scores for the three described kernels on all four data sets, both for atom-bond graphs and for augmented molecular graphs. The standard deviation over the ten folds is also shown. Note that the number of folds influences the height of the reported ROC_{50}, since a higher number of folds means smaller test sets, while the cutoff number of false positives stays at 50. Indeed, for PTC, the ROC_{50} for each of the 10 folds is equal to the AUROC due to the small number of instances.

The answer to **Q**1 is clear from Table 2: augmentation was substantially beneficial with regard to generalisation capacity. All tested tasks and all kernels benefit from augmentation. When computing the *relative error reduction* (RER= $\frac{e-e'}{e}$ where e is the error of the original method and e' is the error of the novel method[6]) we observe an average reduction of 20% for the WDK and ELSPK (with a remarkable 45% error reduction when the ELSPK is used on the Bursi data set) while the PMCSK presents a more modest 5% average error reduction. Note that diminishing marginal returns are to be expected as performance increases. The ROC_{50} results confirm that the augmen-

[6]Note that we have here abused the notation and consider $e = 1-$AUROC rather than $e = 1-$accuracy

Table 2: Predictive performance of the three kernels on unaugmented and augmented molecular graphs: average and standard deviation of the AUROC and ROC_{50} scores over 10 folds. NCI-60 and PTC numbers are averages over 60 and 4 tasks, respectively. For comparison, the expected ROC_{50} for random predictions is also shown.

	NCI-60 (avg.)	HIV CA vs. CM	HIV CACM vs. CI	HIV CA vs. CI	PTC (avg.)	Bursi
AUROC (%)						
ELSPK(G)	70.2 ± 2.5	77.4 ± 2.5	76.0 ± 2.2	90.1 ± 4.5	59.8 ± 8.5	76.5 ± 1.2
ELSPK(G_{aug})	74.7 ± 2.5	80.1 ± 5.7	81.3 ± 2.3	92.8 ± 2.5	64.7 ± 9.5	87.0 ± 1.5
WDK(G)	73.3 ± 2.5	77.7 ± 5.8	80.5 ± 2.3	92.5 ± 3.6	62.3 ± 10	83.6 ± 1.5
WDK(G_{aug})	77.9 ± 2.4	82.0 ± 4.8	83.4 ± 2.2	94.5 ± 2.9	66.1 ± 9.7	87.6 ± 1.1
PMCSK(G)	79.6 ± 2.2	82.6 ± 6.2	81.8 ± 2.2	93.0 ± 3.7	64.5 ± 8.8	90.5 ± 1.3
PMCSK(G'_{aug})	80.3 ± 2.2	82.8 ± 6.2	83.2 ± 2.1	93.4 ± 3.4	65.6 ± 8.8	91.5 ± 1.1
ROC_{50} (%)						
$\mathbb{E}(ROC_{50})$	15.7	23.1	0.6	0.6	50.0	12.9
ELSPK(G)	37.3 ± 4.5	60.5 ± 6.0	5.7 ± 2.2	23.9 ± 4.6	59.8 ± 8.5	45.3 ± 2.9
ELSPK(G_{aug})	42.0 ± 4.8	64.4 ± 7.9	13.5 ± 3.5	39.1 ± 6.5	64.7 ± 9.5	63.8 ± 3.7
WDK(G)	40.2 ± 4.4	60.5 ± 9.1	10.5 ± 2.1	29.7 ± 6.6	62.3 ± 10	58.6 ± 3.5
WDK(G_{aug})	47.1 ± 4.7	68.8 ± 7.6	19.9 ± 2.6	59.2 ± 6.0	66.1 ± 9.7	67.0 ± 3.9
PMCSK(G)	51.7 ± 4.5	71.5 ± 9.4	34.2 ± 3.3	66.0 ± 6.3	64.5 ± 8.8	72.4 ± 3.7
PMCSK(G'_{aug})	52.8 ± 4.5	72.2 ± 9.6	35.5 ± 3.1	67.7 ± 6.9	65.6 ± 8.8	74.8 ± 3.5

tation is also effective in increasing the performance for what are considered the most trustworthy predictions by the WDK and ELSPK methods with an average RER of 17% (4% for the PMCSK).

Note that the WDK results reported in Table 2 are worse than those obtained by Ceroni et al. since we use the basic WDK and did not implement all refinements of Menchetti et al.

Due to space and time constraints, there obviously remain a large number of different graph kernel approaches, such as,[37–39] for which we do not obtain direct experimental evidence whether the augmentation procedure presented in this paper leads to significant predictive performance increase. As a general remark, we note that the augmentation affects several key characteristics of the input graphs, for example on the Bursi data set we observe the following changes: the vertex and edge label alphabet is increased from 13 to 69 and from 4 to 9 respectively; the vertex degree distribution and the vertex and edge count distribution changes as shown in Table 3. For some graph kernels these differences (increased average label alphabet and degree size, and number of vertices and edges) can lead to a significant increase in the expected runtime, a negative aspect

Table 3: Descriptive statistics of unaugmented and augmented molecular graphs for the Bursi data set

	Mean	Min	Quartile 1	Quartile 2	Quartile 3	Max
deg_{V_G}	2.12	0	2	2	3	4
$deg_{V_{G_{aug}}}$	3.78	0	3	3	4	81
$\|V_G\|$	16.9	2	11	16	21	214
$\|V_{G_{aug}}\|$	23.08	4	15	21	28	294
$\|E_G\|$	17.88	1	11	17	23	217
$\|E_{G_{aug}}\|$	43.61	4	25	39	56	542

that has to be weighted against the expected performance increase. For example, both the method proposed by Riesen and Bunke [38] and the one proposed by Rupp et al.,[39] make use of the Kuhn-Munkres assignment algorithm[40] that has a $O(|V|^3)$ complexity. The method from Horváth et al. [41] instead, counts the number of cycles in a graph and can suffer from the many cycles introduced by the *part-of* links added by the augmentation procedure. Here, as in the PMCSK case, a possible workaround would be to eliminate such links and consider the moiety graph as a disconnected component w.r.t. the original chemical graph.

For the three types of kernels that we have selected, we observe only a modest runtime overhead: 1.5 times for ELSPK and WDK and negligible for PMCSK. This latter result can be explained by the small size of the moiety graphs (the number of vertices/edges is $\approx 1/3$ than for the standard molecular graphs). As a consequence, the additional runtime spent by the PMCSK algorithm on the moiety graphs is two orders of magnitude lower than the time spent for the standard molecular graphs.

In Table 4 we report the runtime (in seconds) required for the augmentation pre-processing step compared to the actual kernel computation. Obviously, the augmentation step is of linear complexity in the number of molecules, while the Gram matrix computation is quadratic. The throughput on the large HIV data set of the latter, dominant step is about 430,000 molecule-molecule comparisons per second for ELSPK, 8,000 for WDK, and just 70 for PMCSK. The programs were executed on an Intel Core2 Quad Q9550 CPU (2.83GHz), except for the HIV data set which was run on an Intel Xeon E5420 CPU (2.5GHz) due to 64-bit support of the operating system. The

Table 4: Runtime cost of graph augmentation: CPU time in seconds to calculate the Gram matrix for each kernel, without and with augmentation. The time required for augmentation itself is indicated separately. (*) For the PMCSK on the HIV data, the subgraphs were derived only from the 1503 confirmed active or moderately active molecules.

	NCI-60	HIV	PTC	Bursi
Number of molecules	3910	42687	417	4337
Augmentation time	$3.5 \cdot 10^2$	$3.4 \cdot 10^3$	$3.4 \cdot 10^1$	$1.2 \cdot 10^2$
ELSPK(G)	$4.2 \cdot 10^1$	$3.9 \cdot 10^3$	$1.0 \cdot 10^0$	$3.6 \cdot 10^1$
ELSPK(G_{aug})	$7.7 \cdot 10^1$	$4.2 \cdot 10^3$	$2.0 \cdot 10^0$	$5.7 \cdot 10^1$
WDK(G)	$1.8 \cdot 10^3$	$1.6 \cdot 10^5$	$8.0 \cdot 10^0$	$1.1 \cdot 10^3$
WDK(G_{aug})	$2.3 \cdot 10^3$	$2.3 \cdot 10^5$	$1.4 \cdot 10^1$	$1.5 \cdot 10^3$
PMCSK(G)	$2.8 \cdot 10^5$	$3.3 \cdot 10^4$ (*)	$6.2 \cdot 10^2$	$3.5 \cdot 10^5$
PMCSK(G'_{aug})	$2.8 \cdot 10^5$	$3.3 \cdot 10^4$ (*)	$6.3 \cdot 10^2$	$3.5 \cdot 10^5$

programs are all essentially single-threaded. The table shows net consumed CPU time, except for the augmentation process where we were only able to measure wall clock time. The time for the ELSPK is for our own implementation; its performance characteristics may bear no resemblance to Borgwardt and Kriegel's original version.

Q2 *How do augmented graph kernels compare to the current QSAR state-of-the-art?*

The current best published kernel for the NCI-60 data sets is from Wale et al. [2] For comparison with the state-of-the-art, we used their Graph Fragments (GF) kernel on NCI-60 on the same cross-validation folds as for the other kernels. Graph Fragments, as it is implemented by its authors, is not a general graph kernel, but is highly specialized for molecular atom-bond graphs. For example, one of the built-in constraints is that nodes cannot have a degree larger than 5. This prevented us from using the kernel on the augmented graphs, or even the separate moiety graphs as for the PMCSK. Furthermore, we did not implement the length-differentiated min-max kernel, but rather used the descriptors produced by the AFGen program in the same experimental settings as for all other kernels.

According to the binomial sign test at the 5% level, ELSPK performed worse than GF on all cell lines except one, where there was no significant difference. WDK showed 35 draws and 25 losses. PMCSK, however, performed significantly better than GF on 28 cell lines and worse in none.

WDK[23] is the method with the highest reported AUROC (84.2%) for the CA vs. CM task in HIV. Swamidass et al. [42] recently reported an AUROC of 84.5% on the CA+CM vs. CI task. Wale et al. report an AUROC of 95.0% for the CA vs. CI task, using Acyclic Fragments.

Wale et al. also report an AUROC of 71.1% for PTC using Path Fragments and 71.0% using GF.

To our knowledge, the highest AUROC on the Bursi data set was obtained by Saigo et al. who report a best AUROC of 88.9% using gBoost,[43] and an accuracy of 82.5%. Kazius et al. [33] achieved a training accuracy of 83% with a manually constructed model, using all data without cross-validation. However, they do use a smaller, separate test set, where the model counterintuitively achieves a slightly higher accuracy of 85%. They report that the average interlaboratory reproducibility error is 15%, which provides an approximate upper bound on the achievable accuracy. The accuracy obtained with PMCSK(G'_{aug}) was 85%, equal to the estimated upper bound.

Finally, the answer to **Q**2 is that the PMCSK operating on the augmented molecular graphs exhibits a predictive performance that is competitive with state-of-the-art results, in particular on the NCI-60 and Bursi data sets. The predictive power of the other kernels (ELSPK and WDK) is also much improved when working with the augmented graphs. The performance gap with respect to more complex and expensive kernels is significantly reduced.

Conclusions

The major contribution of this paper is the introduction of a simple but effective way to incorporate background knowledge in graph-based representations of molecular data. To demonstrate the effectiveness of the proposed approach, we tested several graph kernel models on the augmented representations. Moreover, the proposed technique has been tested on a wide range of different types of chemoinformatics classification tasks. In all cases we observe a consistent improvement of the predictive performance. Finally, when providing the background knowledge to the PMCSK, we found that it significantly outperforms the current state-of-the-art algorithm on the NCI-60 data

set in 28 of the 60 cell lines, with the other 32 cell lines showing no significant difference in accuracy, and it obtains near-optimal results on the Bursi mutagenicity task.

Supporting information

The software for augmentation, including DMax Chemistry Assistant, is available at `http://dtai.cs.kuleuven.be/dmax/`.

Acknowledgement

The authors thank Leander Schietgat for providing assistance for his PMCSK implementation[24] and for providing the Graph Fragments predictions. The authors are funded by GOA/08/008 "Probabilistic Logic Learning".

References

(1) Lengauer, T.; Lemmen, C.; Rarey, M.; Zimmermann, M. Novel Technologies for Virtual Screening. *Drug Discovery Today* **2004**, *9*, 27–34.

(2) Wale, N.; Watson, I.; Karypis, G. Comparison of Descriptor Spaces for Chemical Compound Retrieval and Classification. *Knowl. Inf. Syst.* **2008**, *14*, 347–375.

(3) Brown, N. Chemoinformatics—an Introduction for Computer Scientists. *ACM Comput. Surv.* **2009**, *41*, 1–38.

(4) Mezey, P. G. QSAR and the Ultimate Molecular Descriptor: the Shape of Electron Density Clouds. *J. Math. Chem.* **2009**, *45*, 544–549.

(5) Bader, R. F. W.; Popelier, P. L. A.; Keith, T. A. Theoretical Definition of a Functional Group and the Molecular Orbital Paradigm. *Angew. Chem. Int. Edit.* **1994**, *33*, 620–631.

(6) Ando, H. Y.; Dehaspe, L.; Luyten, W.; Craenenbroeck, E. V.; Vandecasteele, H.; Meervelt, L. V. Discovering H-Bonding Rules in Crystals with Inductive Logic Programming. *Mol. Pharm.* **2006**, *3*, 665–674.

(7) De Raedt, L. Representations for Mining and Learning. In *Logical and Relational Learning*; Gabbay, D. M., Siekmann, J., Eds., 1st ed.; Springer-Verlag: Berlin Heidelberg, Germany, 2008; pp 71–114.

(8) Blockeel, H.; Dehaspe, L.; Ramon, J.; Struyf, J.; Van Assche, A.; Vens, C.; Fierens, D. *The ACE Data Mining System: User's Manual*, 2009, `http://dtai.cs.kuleuven.be/ACE/doc/ACEuser-1.2.16.pdf` (accessed July 2, 2009).

(9) Vandecasteele, H.; Van Craenenbroeck, E. *DMax Functional Group and Ring Library*, 2002.

(10) De Raedt, L. An Introduction to Logic. In *Logical and Relational Learning*; Gabbay, D. M., Siekmann, J., Eds., 1st ed.; Springer-Verlag: Berlin Heidelberg, Germany, 2008; pp 17–40.

(11) Takahashi, Y.; Sukekawa, M.; Sasaki, S.-i. Automatic Identification of Molecular Similarity using Reduced-Graph Representation of Chemical Structure. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 639–643.

(12) Gillet, V. J.; Willett, P.; Bradshaw, J. Similarity Searching using Reduced Graphs. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 338–345.

(13) Barker, E. J.; Gardiner, E. J.; Gillet, V. J.; Kitts, P.; Morris, J. Further Development of Reduced Graphs for Identifying Bioactive Compounds. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 346–356.

(14) Stiefl, N.; Watson, I. A.; Baumann, K.; Zaliani, A. ErG: 2D Pharmacophore Descriptions for Scaffold Hopping. *J. Chem. Inf. Model.* **2006**, *46*, 208–220.

(15) Rarey, M.; Dixon, S. J. Feature Trees: A New Molecular Similarity Measure based on Tree Matching. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 471–490.

(16) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and other Kernel Based Methods*; Cambridge University Press: Cambridge, UK, 2000.

(17) Schölkopf, B.; Smola, A. *Learning with Kernels*; MIT Press: Cambridge, MA, 2002.

(18) Gärtner, T. A Survey of Kernels for Structured Data. *SIGKDD Explor.* **2003**, *5*, 49–58.

(19) Borgwardt, K.; Kriegel, H. Shortest-Path Kernels on Graphs. In *Proceedings of the 5th IEEE International Conference on Data Mining*; Wu, X., Ed.; IEEE Computer Society: Houston, Texas, 2005; Vol. 5, pp 74–81.

(20) Floyd, R. W. Algorithm 97, Shortest Path. *Commun. ACM* **1962**, *5*, 345.

(21) Warshall, S. A Theorem on Boolean Matrices. *J. ACM* **1962**, *9*, 11–12.

(22) Haussler, D. *Convolution Kernels on Discrete Structures*; 1999.

(23) Menchetti, S.; Costa, F.; Frasconi, P. Weighted Decomposition Kernels. In *Proceedings of the 22nd International Conference on Machine Learning*; De Raedt, L., Wrobel, S., Eds.; ACM: New York, NY, 2005; Vol. 119, pp 585–592.

(24) Schietgat, L.; Costa, F.; Ramon, J.; De Raedt, L. Maximum Common Subgraph Mining: a Fast and Effective Approach Towards Feature Generation. In *Proceedings of the 7th International Workshop on Mining and Learning with Graphs*; Blockeel, H., Borgwardt, K., Yan, X., Eds.; 2009; Vol. 7, pp 1–3.

(25) Willett, P. Similarity-based Virtual Screening using 2D Fingerprints. *Drug Discovery Today* **2006**, *11*, 1046–1051.

(26) Gower, J. A General Coefficient of Similarity and Some of its Properties. *Biometrics* **1971**, *27*, 857–871.

(27) Schietgat, L.; Ramon, J.; Bruynooghe, M.; Blockeel, H. An Efficiently Computable Graph-based Metric for the Classification of Small Molecules. In *Proceedings of the 11th Interna-*

tional Conference on Discovery Science; Boulicaut, J.-F., Berthold, M. R., Horváth, T., Eds.; Springer-Verlag: Berlin Heidelberg, Germany, 2008; Vol. 5255, pp 197–209.

(28) Swamidass, S. J.; Chen, J.; Bruand, J.; Phung, P.; Ralaivola, L.; Baldi, P. Kernels for Small Molecules and the Prediction of Mutagenicity, Toxicity and Anti-Cancer Activity. *Bioinformatics* **2005**, *21*, i359–368.

(29) Kramer, S.; De Raedt, L.; Helma, C. Molecular Feature Mining in HIV Data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*; Lee, D., Ed.; ACM Press: New York, NY, 2001; pp 136–143.

(30) Ceroni, A.; Costa, F.; Frasconi, P. Classification of Small Molecules by Two- and Three-Dimensional Decomposition Kernels. *Bioinformatics* **2007**, *23*, 2038–2045.

(31) Toivonen, H.; Srinivasan, A.; King, R. D.; Kramer, S.; Helma, C. Statistical Evaluation of the Predictive Toxicology Challenge 2000-2001. *Bioinformatics* **2003**, *19*, 1183–1193.

(32) Benigni, R.; Giuliani, A. Putting the Predictive Toxicology Challenge into Perspective: Reflections on the Results. *Bioinformatics* **2003**, *19*, 1194–1200.

(33) Kazius, J.; McGuire, R.; Bursi, R. Derivation and Validation of Toxicophores for Mutagenicity Prediction. *J. Med. Chem.* **2005**, *48*, 312–320.

(34) Gribskov, M.; Robinson, N. L. Use of Receiver Operating Characteristic (ROC) Analysis to Evaluate Sequence Matching. *Comput. Chem.* **1996**, *20*, 25–33.

(35) Joachims, T. Making Large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*; Scholkopf, B., Burges, C., Smola, A., Eds.; MIT-Press: Cambridge, MA, 1999.

(36) Morik, K.; Brockhausen, P.; Joachims, T. Combining Statistical Learning with a Knowledge-based Approach – a Case Study in Intensive Care Monitoring. In *Proceedings of the 16th*

International Conference on Machine Learning (ICML-99); Bratko, I., Džeroski, S., Eds.; Morgan Kaufmann: San Fransisco, CA, 1999; pp 268–277.

(37) Gärtner, T.; Horváth, T.; Le, Q. V.; Somla, A. J.; Wrobel, S. Kernel Methods for Graphs. In *Mining Graph Data*; Cook, D. J., Holder, L. B., Eds.; John Wiley and Sons: Hoboken, New Jersey, 2007.

(38) Riesen, K.; Bunke, H. Reducing the Dimensionality of Dissimilarity Space Embedding Graph Kernels. *Eng. Appl. Artif. Intel.* **2009**, *22*, 48–56.

(39) Rupp, M.; Proschak, E.; Scheider, G. Kernel Approach to Molecular Similarity based on Iterative Graph Similarity. *J. Chem. Inf. Model.* **2007**, *47*, 2280–2286.

(40) Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Indust. and Appl. Math.* **1957**, *5*, 32–38.

(41) Horváth, T.; Gärtner, T.; Wrobel, S. Cyclic Pattern Kernels for Predictive Graph Mining. In *International Conference on Knowledge Discovery and Data Mining, Proceedings of the Tenth ACM SIGKDD conference (KDD2004)*; Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W., Eds.; ACM: New York, NY, 2004.

(42) Swamidass, S. J.; Azencott, C.-A.; Lin, T.-W.; Gramajo, H.; Tsai, S.-C.; Baldi, P. Influence Relevance Voting: an Accurate And Interpretable Virtual High Throughput Screening Method. *J. Chem. Inf. Model.* **2009**, *49*, 756–766.

(43) Saigo, H.; Nowozin, S.; Kadowaki, T.; Kudo, T.; Tsuda, K. gBoost: a Mathematical Programming Approach to Graph Classification and Regression. *Mach. Learn.* **2009**, *75*, 69–89.